Journal of Mathematics and Informatics
Vol. 7, 2017, 37-44
ISSN: 2349-0632 (P), 2349-0640 (online)
Published 23 March 2017
www.researchmathsci.org
DOI: http://dx.doi.org/10.22457/jmi.v7a5

Journal of
Mathematics and
Informatics

Picture Array Generation Based on Membrane Systems and 2D Context-Free Grammars

P.S. Azeezun Nisha[1], S. Hemalatha[2], N.Gnanamalar David[3] and K.G. Subramanian[4]

[1]Department of Mathematics, J.B.A.S. College for Women, Chennai, India
[2]Department of Mathematics, S.D.N.B Vaishnav College for Women, Chennai, India
[2]Department of Mathematics, Madras Christian College, Chennai, India
Email: kgsmani1948@gmail.com

Received 14 March 2017; accepted 20 March 2017

Abstract. The theory of automata and formal languages is an important branch of theoretical computer science. In two-dimensional (2D) language theory, extended 2D context free picture grammar (E2DCFPG) constitutes a theoretical model of picture array generation making use of nonterminal symbols and groups of context-free string grammar rules in rewriting a column or row of nonterminal symbols in a picture array with no priority in the rewriting. Here a bio-inspired model of P system in the framework of membrane computing is proposed by involving the E2DCFPG kind of rules in the regions of the P system in generating picture arrays. The generative power of the resulting model of array P system is examined by comparing with some well-known 2D grammar models.

Keywords: Two-dimensional array; context-free grammar; membrane computing

1. Introduction

The theory of formal languages and automata is considered to be the backbone of theoretical computer science. Motivated, by various application problems in the areas of image analysis, picture processing, pattern recognition and several others (see, for example, [2,4,9,12]), two-dimensional language theory was developed as an extension of formal string language theory with different models of picture array generation having been proposed and investigated. In [10], a simple yet general enough two-dimensional picture array generating model, known as Pure 2D context-free grammar (P2DCFG) was introduced. In generating a picture array language L, this 2D grammar involves only terminal symbols and rewrites at a derivation step, all the symbols in a column or row of a picture array making use of a finite set of context-free string grammar rules, thereby yielding the picture arrays of L. There has been other investigation [1] on the mathematical properties of the picture language family of P2DCFG. An extension to this 2D grammar model, called extended 2D context-free picture grammar (E2DCFPG), with more picture generative power than the P2DCFPG was proposed in [11] by allowing variables in the rules of this picture grammar model and collecting the picture arrays generated over a set of terminal symbols.

On the other hand, a bio-inspired computing model, initially called membrane system but later named as P system (in honour of its originator), was introduced by Păun [3] inspired by the structure and functioning of living cells. Among different roles of P systems, rewriting P systems constitute a specific kind wherein finite strings over an alphabet are the objects and context-free rewriting rules are the evolution rules. Extending the rewriting P systems to arrays several P system models for picture array generation, have been considered in the literature (see for example [9]). Here we consider an array P system with array objects and E2DCFPG kind of tables of rules in its regions. We examine the generative power of this array P system model by comparing it with some well-established 2D picture generating models.

2. Basic definitions

For notions of formal language theory we refer to [5] and to [2,4,10] for array grammars. Let *V* be an alphabet, which is a finite set of symbols. A word or a string *u* over *V* is of the form $u_1u_2 \dots u_n$, where $u_i \in V, 1 \le i \le n,$ for some $n \ge 1.$ The length of a word *u* is denoted by |*u*|. The empty word with no symbols is denoted by λ. The set of all words (also called horizontal words) over *V* including the empty word, is denoted by V^*.

For any word $u = u_1u_2 \dots u_n$, the vertical word ${}^t u$ is given by ${}^t u = \begin{matrix} u_1 \\ \vdots \\ u_n \end{matrix}$. We also define ${}^t\left({}^t u\right) = u$ and ${}^t\lambda = \lambda.$ A $m \times n$ picture array (also called an array) α over an alphabet *V* is of the form $M = \begin{matrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{matrix}$ where $a_{ij} \in V, 1 \le i \le m, 1 \le j \le n.$ The set of all picture arrays over *V* is denoted by $V^{**},$ which includes the empty array also denoted by λ. We denote $V^{**} - \lambda$ by V^{++}.

We informally mention some of the picture generating models that are needed in the subsequent section. In the 2D matrix grammar model introduced in [6], called context-sensitive matrix grammar (CSMG), there are two phases of derivation. In the first phase horizontal words over terminal symbols of this phase and referred to as intermediate symbols, is generated by a Chomsky context-sensitive grammar. Then in the second phase, from each of the intermediate symbols in such horizontal words, vertical words of the same length over terminal symbols (of the second phase) are derived to constitute the columns of a picture array over terminal symbols. The derivation in the second phase takes place in parallel with all the rules used in a step being the right-linear rules. We denote the generated picture language class by CSML.

In the first phase of this 2D grammar model, if a Chomsky context-free grammar or a regular grammar is used instead of a context-sensitive grammar, then it is respectively called a context-free matrix grammar (CFMG) or regular matrix grammar (RMG) and the corresponding picture language classes are respectively denoted by CFML and RML.

Extending the 2D matrix grammar, Tabled matrix grammar was introduced in [7] by specifying a finite set of tables of rules in the second phase of generation with each table having either right-linear nonterminal rules or right-linear terminal rules. The rules in a single table are used at a step of derivation in the second phase for generating the columns in parallel. The resulting families of picture array languages are denoted by TCSML, TCFML and TRML. It has been shown that XML $\subset$ TXML for X $\in$ {R, CF, CS }.

We now recall extended 2D context-free picture grammar considered in [11]. In an E2DCFPG, unlike 2D matrix grammar, there is single phase of derivation. Starting with an axiom picture array, either a column of nonterminals or a row of nonterminals present in the array is rewritten respectively by a column table or a row table of context-free rules at a step of derivation. The right sides of all the rules in a table are required to be of equal length to ensure that the rectangular form of the picture array rewritten is maintained. When the derivation terminates yielding a picture array over terminal symbols, the generated array is collected in the language.

Definition 2.1. [11] An extended 2D context-free picture grammar (E2DCFPG) is a 5-tuple $G=(V,T,P_c,P_r,\mathrm{M})$ where *V* is a finite set of symbols ; The elements of $V-T$ are called variables; $T\subset V$ is the set of terminal symbols; $P_c=\{t_{c_i}\,|1\le i\le m\}$, $P_r=\{t_{r_j}\,|1\le j\le n\}$; Each $t_{c_i}, 1\le i\le m,$ called a column table, is a set of context-free rules of the form $A\to\alpha, A\in V-T,\ \alpha\in V^*$ such that for any two rules $A\to\alpha, B\to\beta$ in t_{c_i}, we have $|\alpha|=|\beta|$; Each $t_{r_j}, 1\le j\le n,$ called a row table, is a set of context-free rules of the form $C\to {}^t\gamma, C\in V-T,\ \gamma\in V^*$ such that for any two rules $C\to {}^t\gamma, D\to {}^t\delta$ in t_{r_j}, we have $|\gamma|=|\delta|$; $\mathrm{M}\subset V^{**}-\lambda$ is a finite set of axiom arrays.

Derivations are defined as follows: For any two arrays M_1, M_2, we write $M_1\Rightarrow M_2$, if M_2 is obtained from M_1 by either rewriting every symbol of a column of M_1 by rules of some column table t_{c_i} in P_c or of a row of M_1 by rules of some row table t_{r_j} in P_r. The reflexive transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$. The picture array language $L(G)$ generated by G is the set of picture arrays $\{M\,|\,M_0\Rightarrow^* M\in T^{**},\ \mathit{for\ some}\ M_0\in\mathrm{M}\}$. The family of picture array languages generated by E2DCFPGs is denoted by E2DCFPL.

Example 2.1. Consider the E2DCFPG $G=(V,T,P_c,P_r,\{M_0\})$, where $V=\{S,X,A,B,C,a,b\}$, $T=\{a,b\}$, $P_c=\{t_{c_1},t_{c_2}\}$, $P_r=\{t_{r_1},t_{r_2}\}$, $M_0=\begin{array}{ccc} A & S & A \\ b & X & b \end{array}$; $t_{c_1}=\{S\to ABA, X\to bXb\}$, $t_{c_2}=\{S\to C, X\to a\}$, $t_{r_1}=\left\{A\to \begin{array}{c} A \\ b \end{array}, B\to \begin{array}{c} S \\ X \end{array}\right\}$, $t_{r_2}=\{A\to a, C\to a\}$. An illustration of how the tables of rules are applied is given below. Starting from the axiom array $M_0=\begin{array}{ccc} A & S & A \\ b & X & b \end{array}$ if the sequence of tables of rules applied is $t_{c_1},t_{r_1},t_{c_2},t_{r_2}$, then the successive arrays obtained in the derivation are given by

$$\begin{array}{ccccc} A & A & B & A & A \\ b & b & X & b & b \end{array}, \begin{array}{ccccc} A & A & S & A & A \\ b & b & X & b & b \\ b & b & X & b & b \end{array}, \begin{array}{ccccc} A & A & C & A & A \\ b & b & a & b & b \\ b & b & a & b & b \end{array}, \begin{array}{ccccc} a & a & a & a & a \\ b & b & a & b & b \\ b & b & a & b & b \end{array}$$

with the final array over terminal symbols $\{a,b\}$ being collected in the picture language generated by the E2DCFPG *G*. On interpreting the symbol *b* as blank, the array generate represents the digitized letter *T* over the symbol *a*, of the form

$$\begin{array}{ccccc} a & a & a & a & a \\ & & a & & \\ & & a & & \end{array}$$

3. Array P system with E2DCFPG

We now introduce an array P system involving extended 2D context-free picture grammar rules and rectangular array objects in its regions.

Definition 3.1. An array *P* system (of degree $m\ge 1$) with extended 2D context-free picture grammar rules is a construct $\Pi=(V,T,\mu,F_1,\cdots,F_m,R_1,\cdots,R_m,i_0)$ where *V* is the alphabet, $T\subset V$ is a finite set of terminal symbols; μ is a membrane structure with *m* membranes labelled in a one-to-one manner with $1,2,\cdots,m$; $F_1,\cdots,F_m$ are finite sets of picture arrays over *V* associated with the *m* regions of μ ; $R_1,\cdots,R_m$ are finite sets of column tables or row tables of extended 2D context-free picture grammar rules over *V* (as in a E2DCFPG) associated with the *m* regions of μ; the tables have one of the attached targets: *here, out, in* (in general, the target indication *here* is not mentioned explicitly and is understood); finally, i_0 is the label of an elementary membrane of μ (the output membrane).

A computation in Π is defined in the same way as in an array-rewriting *P* system with the successful computations being the halting ones: each array, from each region of the system, which can be rewritten by a column/row table of rules (as done in a E2DCFPG) associated with that region (membrane), should be rewritten; this means that one table of rules is applied; the array obtained by rewriting is placed in the region indicated by the target associated with the table used (if the attached target is *here*, it means that the array remains in the same region, if the target is *out*, it means that the array exits the current membrane; and if the target is *in*, it means that the array is immediately sent to one of the directly lower membranes, nondeterministically chosen if several exist there; if no internal membrane exists, then a table with the target indication *in* cannot be used.) A computation is successful only if it stops, a configuration is reached where no table of rules can be applied to the existing picture arrays. The result of a halting computation consists of picture arrays over T placed in the membrane with label i_0 in the halting configuration.

The set of all such arrays computed or generated by a system Π is denoted by $AL(\Pi)$. The family of all array languages $AL(\Pi)$ generated by systems Π as above, with at most *m* membranes, is denoted by $AP_m(E2DCFPG)$.

Example 3.1. Consider the array *P* system with two membranes and extended 2D context-free picture grammar rules given by

$\Pi_{abc} = (\{A, B, X, Y, Z, a, b, c\}, \{a, b, c\}, [_1 [_2]_2]_1, F_1, \phi, R_1, R_2, 2)$ where $F_1 = AB$; R_1 consists of column tables t_{c_1}, t_{c_2} each with target *in* and R_2 consists of the column table t_{c_3} with target *out*, the column table t_{c_4} and the row tables t_{r_1}, t_{r_2}. The tables of rules are as follows: $t_{c_1} = \{A \to XAY\}$, $t_{c_2} = \{A \to XY\}$, $t_{c_3} = \{B \to ZB\}$, $t_{c_4} = \{B \to Z\}$, $t_{r_1} = \left\{ X \to \begin{matrix} a \\ X \end{matrix}, Y \to \begin{matrix} b \\ Y \end{matrix}, Z \to \begin{matrix} c \\ Z \end{matrix} \right\}$, $t_{r_2} = \left\{ X \to \begin{matrix} a \\ a \end{matrix}, Y \to \begin{matrix} b \\ b \end{matrix}, Z \to \begin{matrix} c \\ c \end{matrix} \right\}$.

A computation starts with axiom array AB in region 1. The region 2 initially has no array in it. If rules of table t_{c_1} are applied, then the array $X\ AY\ B$ produced is sent into the inner region 2 due to the target indication *in* of the table t_{c_1}. Here if the rules of table t_{c_3} are applied, then the array $X\ AY\ B$ produced is sent back to region 1, due to the target indication *out* of the table t_{c_3}. The process can repeat. On the other hand application of the table t_{c_2} in region (sending the array to region 2) followed by the application of the table t_{c_4} in region 2 will yield an array in region 2 of the form $X \cdots XY \cdots YZ \cdots Z$. The application of the row table t_{r_1} certain number of times followed by the application of row table t_{r_2} will yield an array over $\{a, b, c\}$ with m $(m \geq 2)$ rows and $3n$ $(n \geq 1)$ columns with the symbols in the first *n* columns over $\{a\}$, the next *n* columns over $\{b\}$ and the last *n* columns over $\{c\}$.

4. Comparison results

We now obtain comparison results on the generative power of array P system with extended 2D CF picture grammar rules and rectangular array objects in its regions.

Theorem 4.1. $E2DCFPL = AP_1(E2DCFPG) \subset AP_2(E2DCFPG)$

Proof: The equality $E2DCFPL = AP_1(E2DCFPG)$ in the statement is straightforward. In fact the array P system with one membrane and E2DCFPG rules will generate exactly the picture array languages in the class E2DCFPL since there is only one membrane and the tables of rules that rewrite arrays are only E2DCFPG kind of tables of rules.

The inclusion $AP_1(E2DCFPG) \subseteq AP_2(E2DCFPG)$ follows from the definition of $AP_m(E2DCFPG)$. In order to prove the proper inclusion, consider the picture language L_1 consisting of $m \times 3n, (m \geq 2,\ n \geq 1)$ picture arrays with the property *p* given by the following: In a $m \times 3n$ picture array, the first *n* columns are over $\{a\}$, the next *n* columns are over $\{b\}$, and the last *n* columns are over $\{c\}$. This language L_1 is in $AP_2(E2DCFPG)$ as shown in Example 3.1 but L_1 cannot be in E2DCFPL. In fact L_1 cannot be generated by any E2DCFPG, as the kind of dependence in the columns given by property *p* cannot be handled by any E2DCFPG.

It is known [6] that $RML \subset CFML \subset CSML$. We now show that the array P system $AP_2(E2DCFPG)$ contains a picture language that cannot be generated by any CSMG.

Theorem 4.2. $AP_2(E2DCFPG) - (CSML - CFML) \neq \phi$.

Proof: We consider the picture array language L_2 consisting of picture arrays, each of which has three arrays of the form M_1, M_2, M_3 with equal number of columns such that M_3 is below M_2 and M_2 is below M_1. Here M_1 is an $(2m+1) \times 3n$, $(m \geq 2, n \geq 1)$ array with the first *n* columns over $\{a\}$, the next *n* columns over $\{b\}$ and the last *n* columns over $\{c\}$. The $(2p+1) \times 3n, (p \geq 2, n \geq 1)$ array M_3 is similar to M_1 having the same number of columns as M_1 but the number of rows of M_3 need not be equal to the number of rows of M_1. The array M_2 with one row and $3n$ columns is given by d^{3n} (with the same *n* defining M_1 or M_3). An array P system with two membranes and E2DCFPG rules that generates L_2 can be constructed by slightly modifying the tables of rules in Example 3.1. The membrane structure $[_1 [_2]_2]_1$, the initial array AB in region 1 are the same as in Example 3.1 . The tables of rules in region 1 are column tables t_{c_1}, t_{c_2} each with target *in* and in region 2, the column table t_{c_3} with target *out*, the column table t_{c_4} and the row table t_{r_1} as in Example 3.1 while the row table t_{r_2} in region 2 is modified as follows: $t_{r_2} = \left\{ X \rightarrow \begin{matrix} d \\ X' \end{matrix}, Y \rightarrow \begin{matrix} d \\ Y' \end{matrix}, Z \rightarrow \begin{matrix} d \\ Z' \end{matrix} \right\}$. In region 2, the following row tables t_{r_3}, t_{r_4} are included where $t_{r_3} = \left\{ X' \rightarrow \begin{matrix} a \\ X' \end{matrix}, Y' \rightarrow \begin{matrix} b \\ Y' \end{matrix}, Z' \rightarrow \begin{matrix} c \\ Z' \end{matrix} \right\}$, $t_{r_3} = \left\{ X' \rightarrow \begin{matrix} a \\ a \end{matrix}, Y' \rightarrow \begin{matrix} b \\ b \end{matrix}, Z' \rightarrow \begin{matrix} c \\ c \end{matrix} \right\}$. The row table t_{r_2} generates a row of *d's* while the additional row table of rules t_{r_3}, t_{r_4} generate the rows below the row of *d's*. Note that the parts of the array above and below the row of *d's* are alike although the number of rows above and below the row of *d's* need not be the same. This picture language L_2 cannot be generated by any CSMG as in any generated array, the feature, namely, a row of *d's* with the arrays above and below it being identical having the same number of rows with each row having *n a's*, followed by *n b's* and *n c's* , cannot be handled by any CSMG.

It is known [7] that $TRML \subset TCFML \subset TCSML.$ We now exhibit a non-trivial picture language which is both in $AP_2(E2DCFPG)$ and TCSML but not in TCFML.

Theorem 4.3. $AP_2(E2DCFPG) \cap (TCSML - TCFML) \neq \phi$.

Proof: We consider the same picture array language L_2 considered in Example 3.1, which shows that $L_2 \in AP_2(E2DCFPG).$ A TCSMG *G* can be constructed to generate L_2. In fact the first phase of the TCSMG *G* will have to generate the well-known [5] context-sensitive language $\{A^n B^n C^n \mid n \geq 1\}$ where A, B, C are the intermediate symbols. In the second phase the following tables of right-linear rules

$t_1 = \{A \rightarrow aA, B \rightarrow bB, C \rightarrow cC\}$, $t_2 = \{A \rightarrow dA', B \rightarrow dB', C \rightarrow dC'\}$,

$t_3 = \{A' \rightarrow aA', B' \rightarrow bB', C' \rightarrow cC'\}$, $t_4 = \{A' \rightarrow a, B' \rightarrow b, C' \rightarrow c\}$ can be included in order to generate the rows of the picture arrays in the language L_2.

5. **Conclusion**

The array P system with tables of context-free rules and E2DCFPG type of rewriting is seen to generate picture languages that cannot be generated by the two-dimensional context-sensitive matrix grammars [2, 6]. Comparisons with other kinds of 2D picture generating models such as the class of recognizable picture languages [2] remain to be explored. Possible application to generation of picture patterns can also be examined as done in [8].

Acknowledgement. The authors thank the reviewers for the useful comments. The author K.G. Subramanian is grateful to UGC, India, for the award of Emeritus Fellowship (No.F.6-6/2016-17/EMERITUS-2015-17-GEN-5933 /(SA-II)) to him to execute his work in the Department of Mathematics, Madras Christian College.

REFERENCES

1. M.M.Bersani, A.Frigeri and A.Cherubini, Expressiveness and complexity of regular pure two-dimensional context-free languages, *Int. J. Comput. Math.,* 90 (2013) 1708-1733.
2. D.Giammarresi and A.Restivo, Two-dimensional languages, In *"Handbook of Formal Languages"* Vol.3, Eds. G. Rozenberg and A. Salomaa, Springer Verlag, 1997, 215 - 267.
3. Gh.Păun, *Computing with Membranes: An Introduction*, Springer-Verlag, Berlin, 2002.
4. A.Rosenfeld, *Picture Languages-Formal Models for Picture Recognition*, Aca-demic Press, New York, 1979.
5. G.Rozenberg, A.Salomaa, (Eds.): *Handbook of Formal Languages* Volumes 1 -3, Springer-Verlag, Berlin, 1997.
6. G.Siromoney, R.Siromoney and K.Krithivasan, Abstract families of matrices and picture languages, *Comput. Graphics Image Process.*, 1 (1972) 234-307.
7. R.Siromoney, K.G.Subramanian and K.Rangarajan, Parallel/Sequential rectangular arrays with tables, *Int. J. Comput. Math.*, 6A (1977) 143-158.
8. G.Samdanielthompson, N.Gnanamalar David and K.G.Subramanian, Alphabetic flat splicing pure context-free grammar systems, *J. Mathematics and Informatics,* 7 (2017) 1-6.
9. K.G.Subramanian, P systems and picture languages, *Lecture Notes in Computer Science*, Springer Verlag, Vol. 4664 (2007) 99-109.
10. K.G.Subramanian, R.M.Ali, M.Geethalakshmi and A.K.Nagar, Pure 2D picture grammars and Languages, *Discrete Appl. Math.,* 157(16) (2009) 3401-3411.
11. K.G.Subramanian, M.Geethalakshmi, A.K.Nagar and S.K.Lee, Two-dimensional picture grammar models, *Proceedings of the 2nd European Modelling Symposium (EMS2008)*, Liverpool Hope university, England, 2008, pp. 263-267.
12. P.S.P.Wang, *Array grammars, Patterns and recognizers,* World Scientific, 1989.